Longitudinal Dampers for Main Injector

Bill Foster , Dennis Nicklaus,
Warren Schappert, Dave Wildman

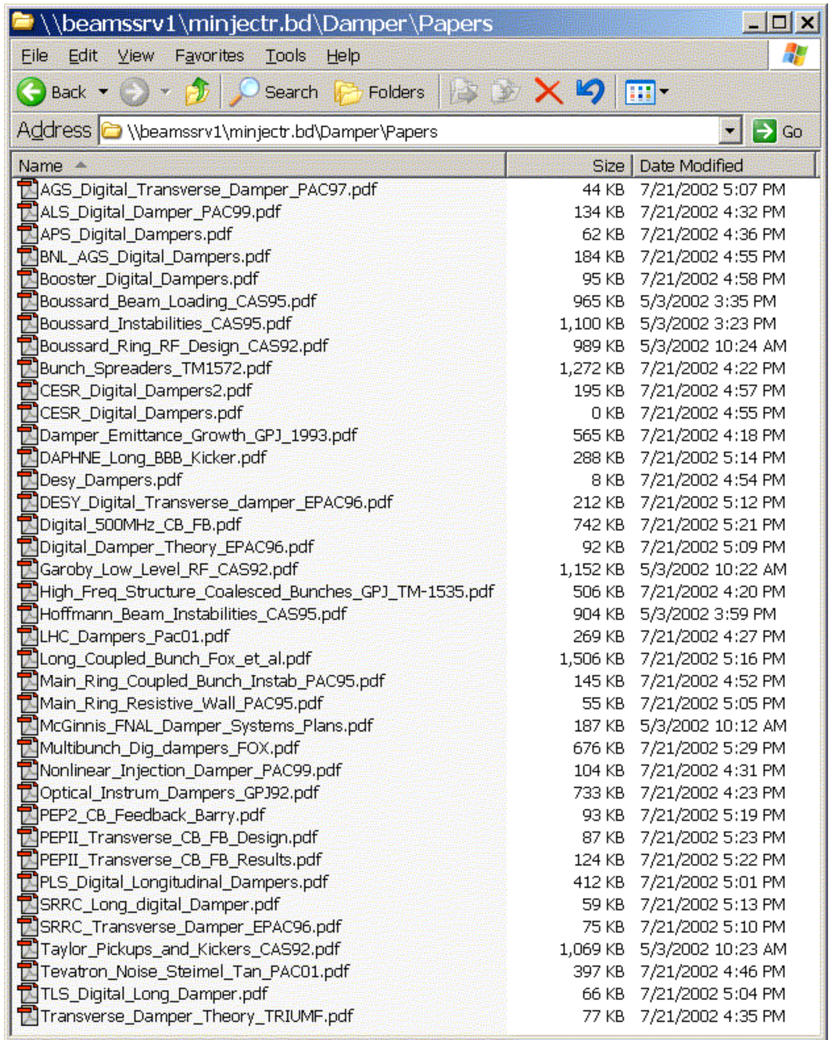
Mar '03

MI/RR Damper

- Documentation
- Hardware (longitudinal)
- ACNET Interface

Mother Lode of Damper Papers:

\\beamssrv1\minjectr.bd\Damper\Papers



- Missing (hardcopy only):
  - Lambertson AIP proceeding on EM theory of pickup & kickers
  - Papers on FNAL super-dampers for MR & TeV
- Intention (~1/4 complete) is to have online directory & summary.

Wide Variety of Beam Dampers Required in MI & Recycler

1) Transverse (X,Y) and Longitudinal

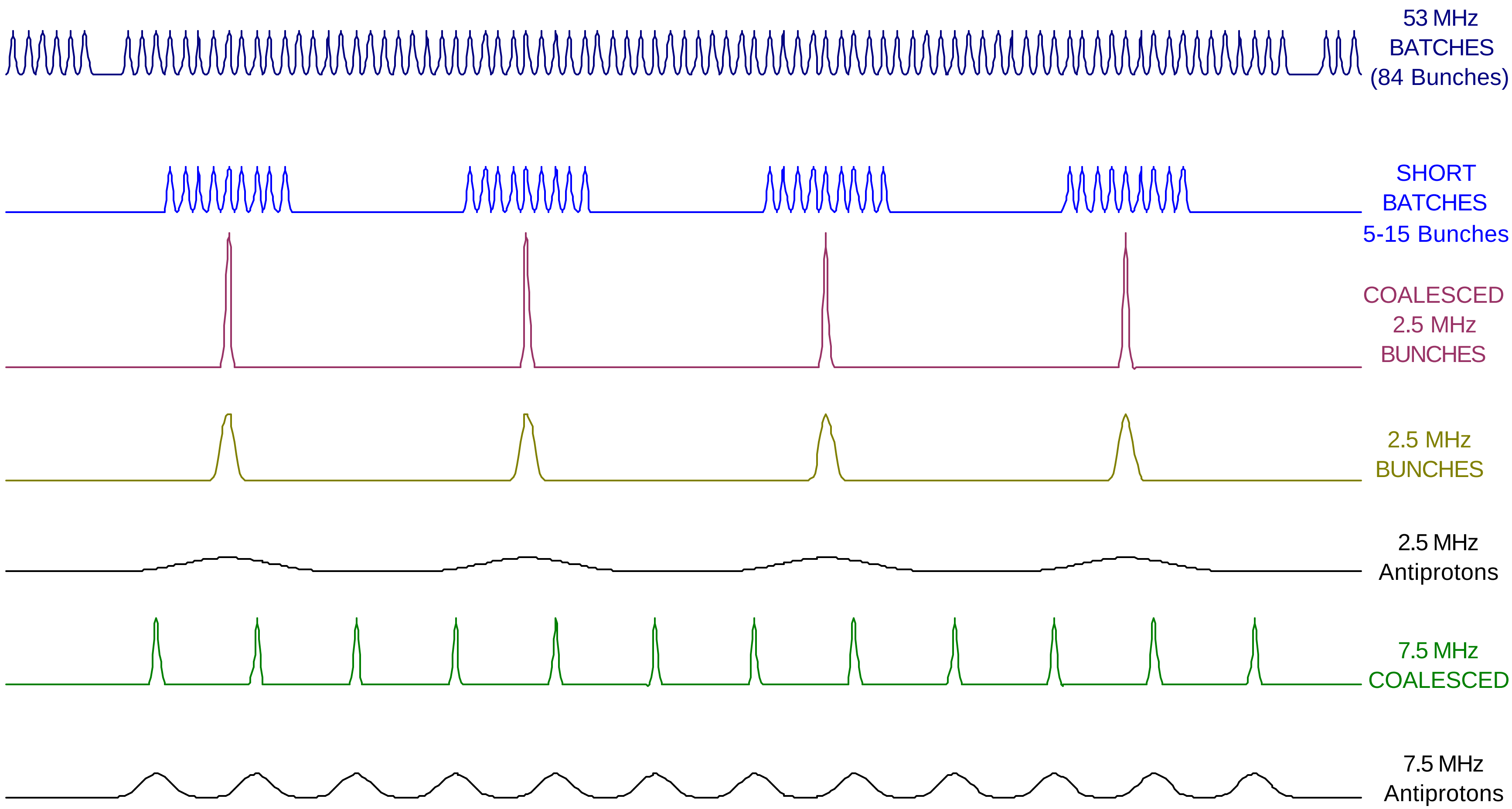
2) 53 MHz, 2.5 MHz, 7.5 MHz, and DC Beam

3) Single Bunches, Full Batches, Short Batches

4) Injection, Ramping, and Stored Beam

5) Pbar and Proton Directions (✍ different timing)

Beam Bunch Structures in Fermilab Main Injector



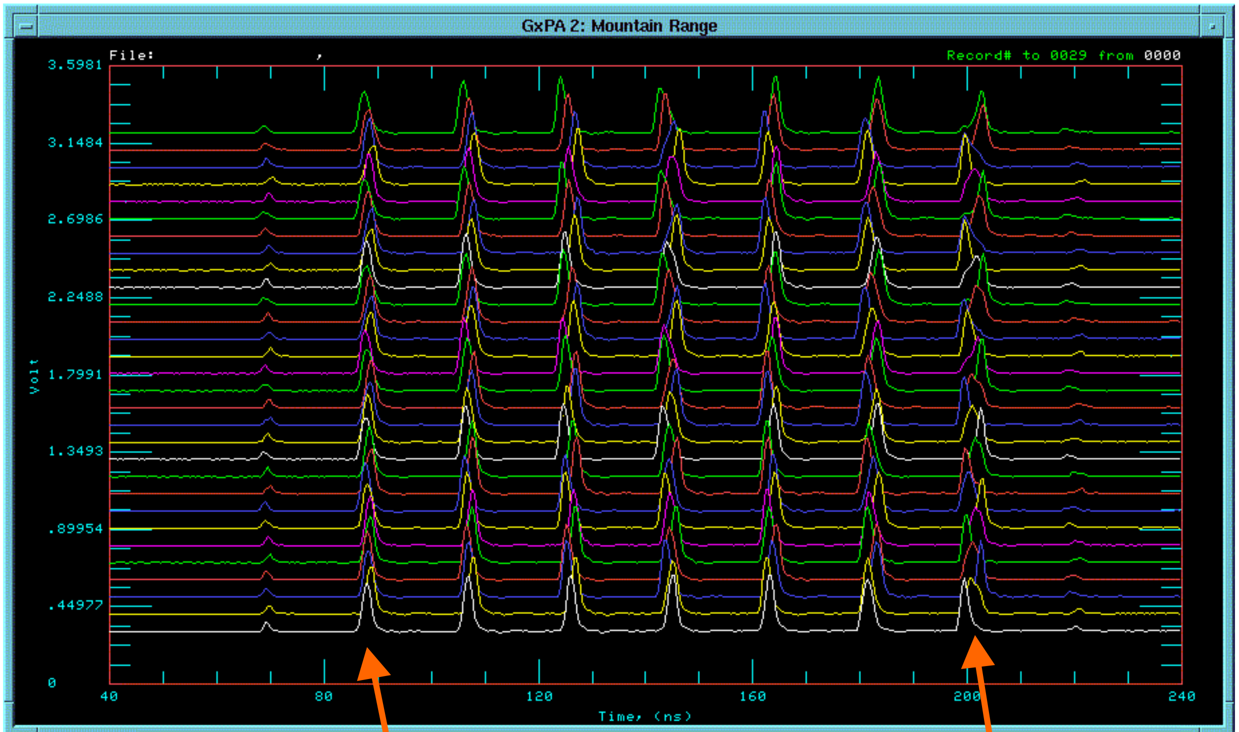
... plus unbunched DC Beam in Recycler...

Damper Operating Modes

	Booster		Main Inj.		Recycler		Tevatron	
	Pbar	P	Pbar	P	Pbar	P	Pbar	P
53 MHz Full Batches		X		X		C		
53 MHz Short Batches			X	X				
53 MHz Coalesced Bunch			X	X			X	X
2.5 MHz Batch (4)			X	C	X	C		
7.5 MHz Batch (12)			X	C				
DC Beam					X	C		

X = Operation   **C** = Commissioning & Tuneup

Longitudinal Beam Instability in MI



- Occurs with as few as 7 bunches (out of 588)
- Prevents low emittance bunch coalescing and efficient Pbar bunch rotation

First Bunch ~ OK

7th Bunch Trashed

see Dave Wildman's Talk

- Driven by cavity wake fields within bunch train
- Seeded by Booster & amplified near MI flat top.

Damper Priorities in Main Injector & Recycler

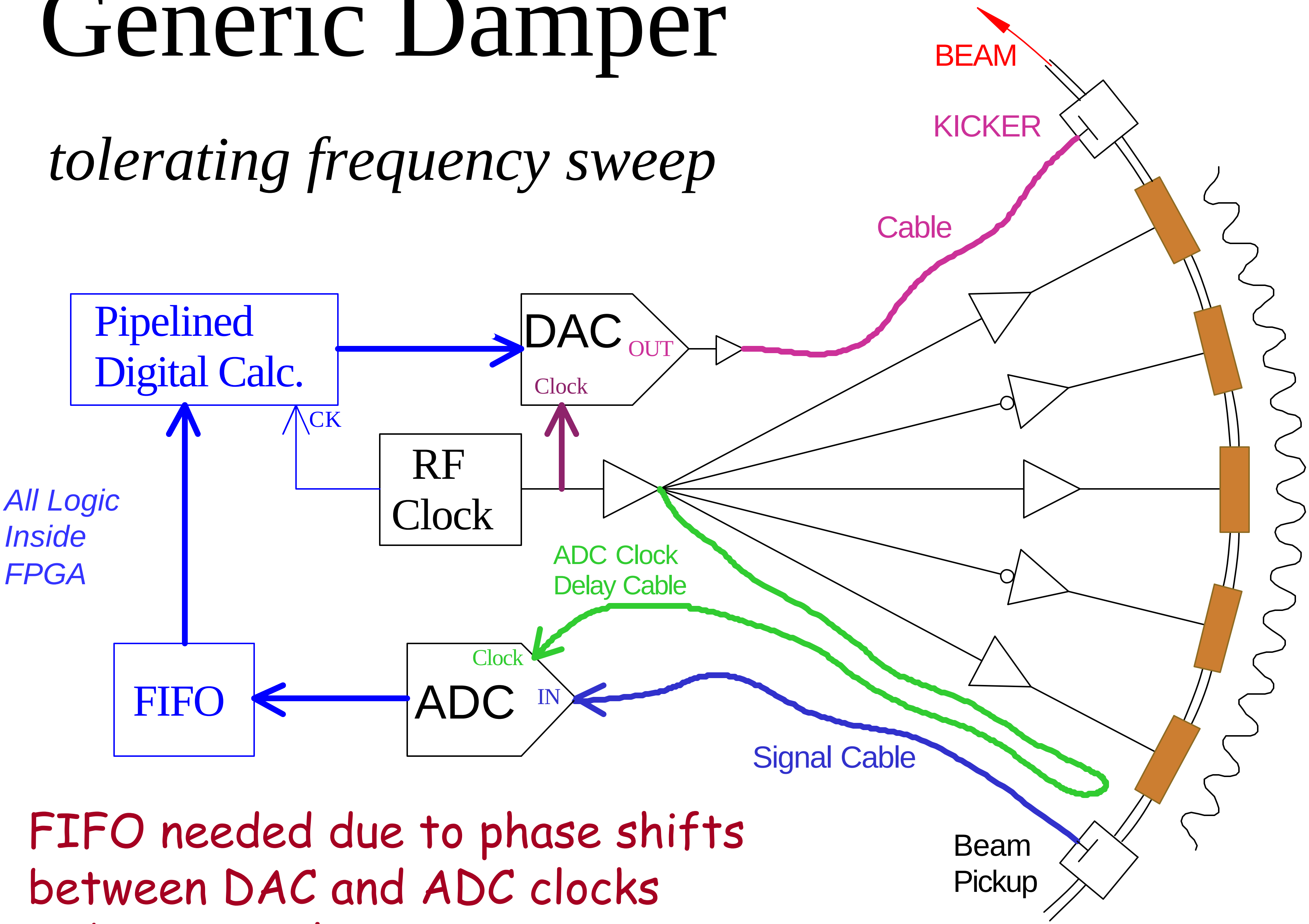
1. Main Injector Longitudinal Dampers
2. Main Injector Transverse Dampers
3. Recycler Transverse Injection Dampers
4. Recycler Longitudinal Dampers
5. Recycler Broadband (DC Beam) Dampers

Advantages of Digital Filters

- Digital filter can also operate at multiple lower frequencies ...simultaneously if desired.
  - ? MI will not be blind for 2.5 and 7.5 MHz Beam
- Digital filters more reproducible (=>spares!)
- Re-use Standard hardware with new FPGA code
  - – or same code with different filter coefficients
- Inputs and Outputs clearly defined (& stored!)
  - – filters can be developed & debugged offline

Generic Damper

tolerating frequency sweep



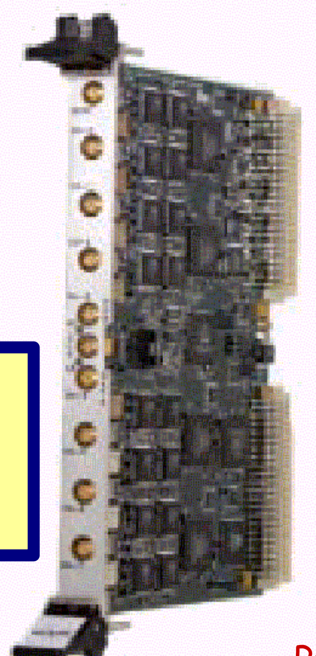
FIFO needed due to phase shifts between DAC and ADC clocks as beam accelerates

Echotek Card Used for Initial Dampers

ECHOTEK CORP.

EIGHT CHANNEL ANALOG TO DIGITAL CONVERTER ~~WITH DIGITAL RECEIVER~~

ECDR-814/X-AD



212 MHz DAC Daughter Card (S. Hansen/ PPD) due this week

FEATURES

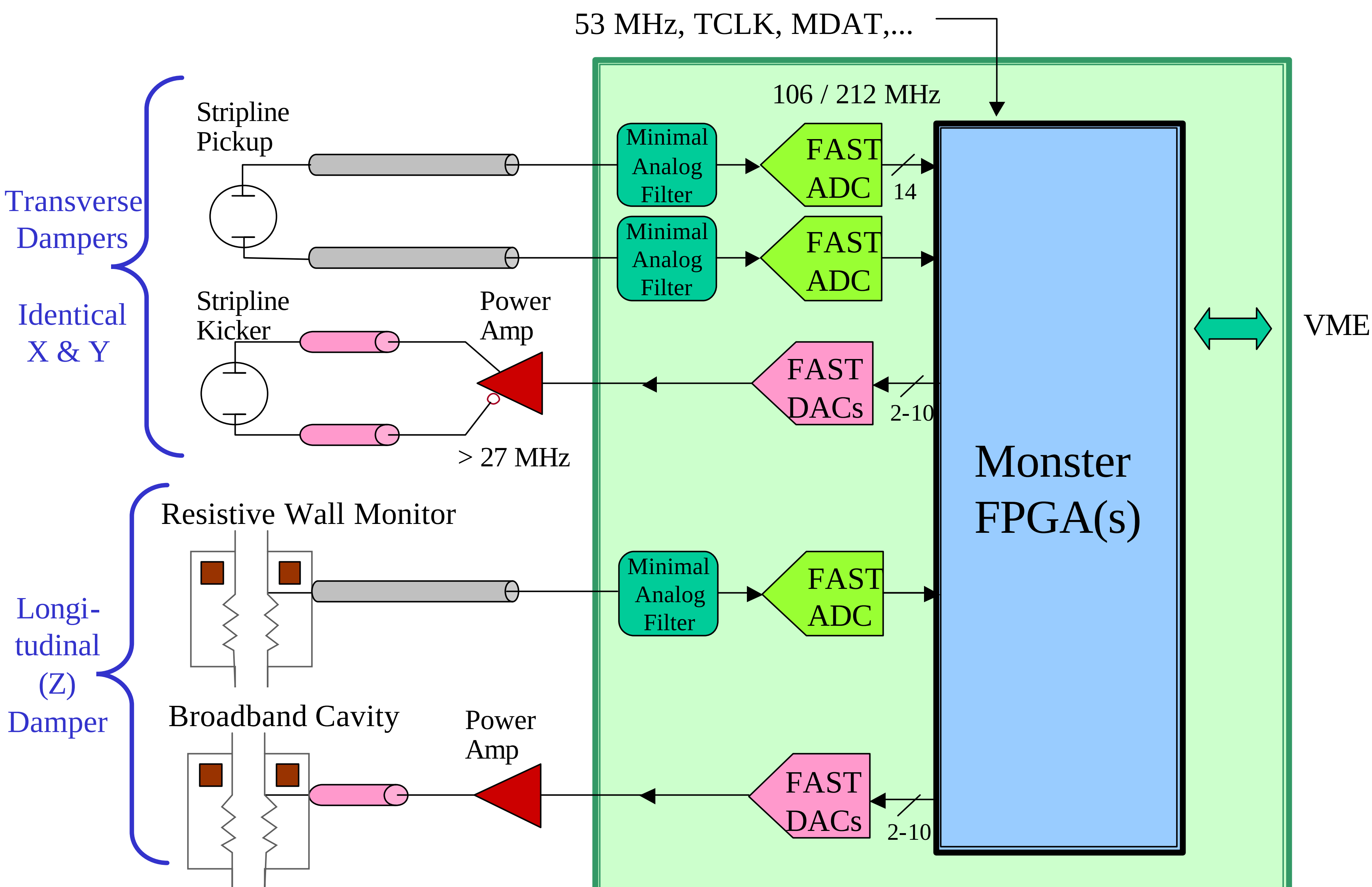
* 8 IF INPUTS
* SIMULTANEOUS SAMPLING
* EIGHT ANALOG TO DIGITAL CONVERTERS (ANALOG DEVICES AD6644, 14 BIT, 65 MSPS) 105 MSPS AD6645
* SFDR > 90 dB FS
* HEADER INSERTION
* VME 64X, SINGLE SLOT
* RACE++ OUTPUT
* AVAILABLE AS A/D CONVERTER AS AN 8, 4, OR 2 CHANNEL MODULE
* VARIABLE GAIN (~ -10 TO +20 dB) OR LOW PASS FILTER

- Prieto, Meyer et. al. evaluating 65MHz DDC for RR BPM upgrade
- Asmanskas, Foster, Schappert testing 105 MHz version for RR Dampers

Digital Signal Processing with FPGA's

- Commercial card from Echotek
  - 8 channels of 14-bit, 106 MHz Digitization
- One card does all dampers for one machine
- Customized FPGA firmware
  - Bill Ashmanskas
  - GW Foster
  - Warren Schappert…
- Handles Wide Variety of Bunch Structure

All-Coordinate Digital Damper



New Damper Board (A. Seminov)

- *SINGLE* high-end FPGA (vs. 5 on Echotek)
- Four 212 MHz ADCs (vs. 106 MHz on Etk.)
- Four 424 MHz DACs (vs. 212 MHz on Etk.)
- Digital Inputs:
  - TCLK, MDAT, BSYNCH, 53 MHz, AA
- Digital Outputs:
  - Pbar/P TTL, scope trigger, 1 GHz serial Links..
- NIM module with Ethernet interface to ACNET

✍ *Other possible uses include replacing entire Booster LLRF system, and Universal BPM.*

Recycler Broadband RF Cavity (3 similar new for broadband damper)

Proceedings of the 1999 Particle Accelerator Conference, New York, 1999

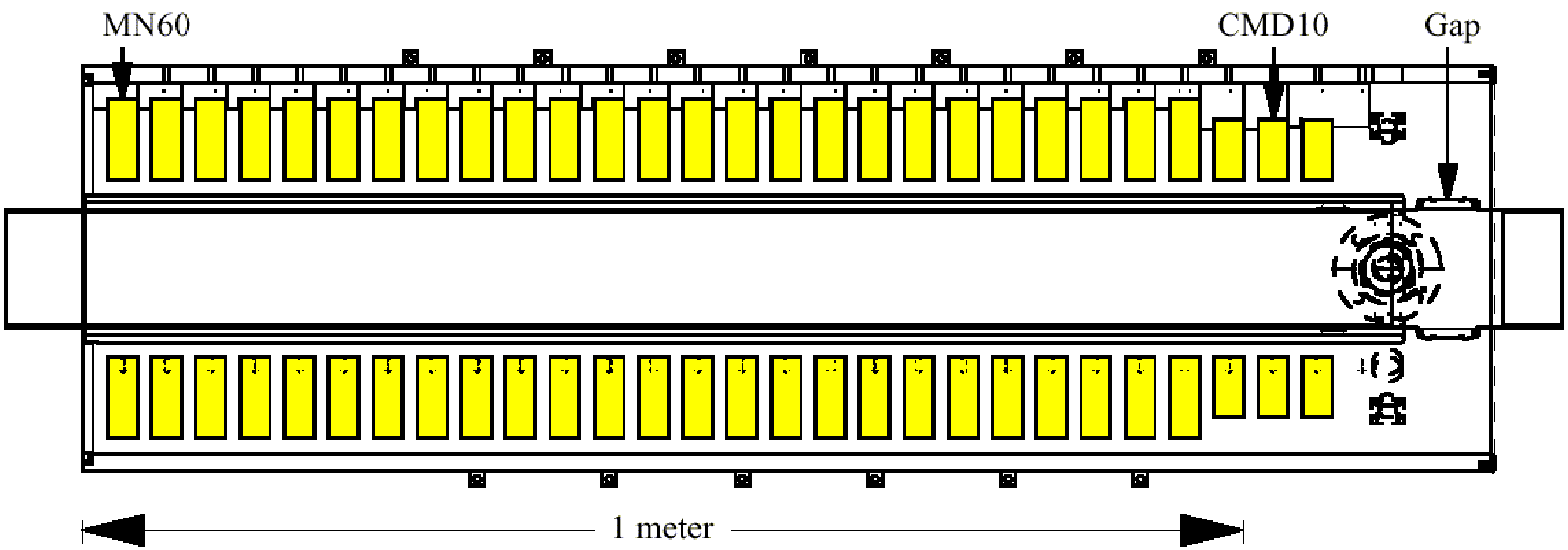


Figure 1: Schematic drawing of Recycler Wideband RF Cavity

Non-Resonant Cavity looks like 50-Ohm Load in parallel with a large Inductor

Wideband Power Amplifiers

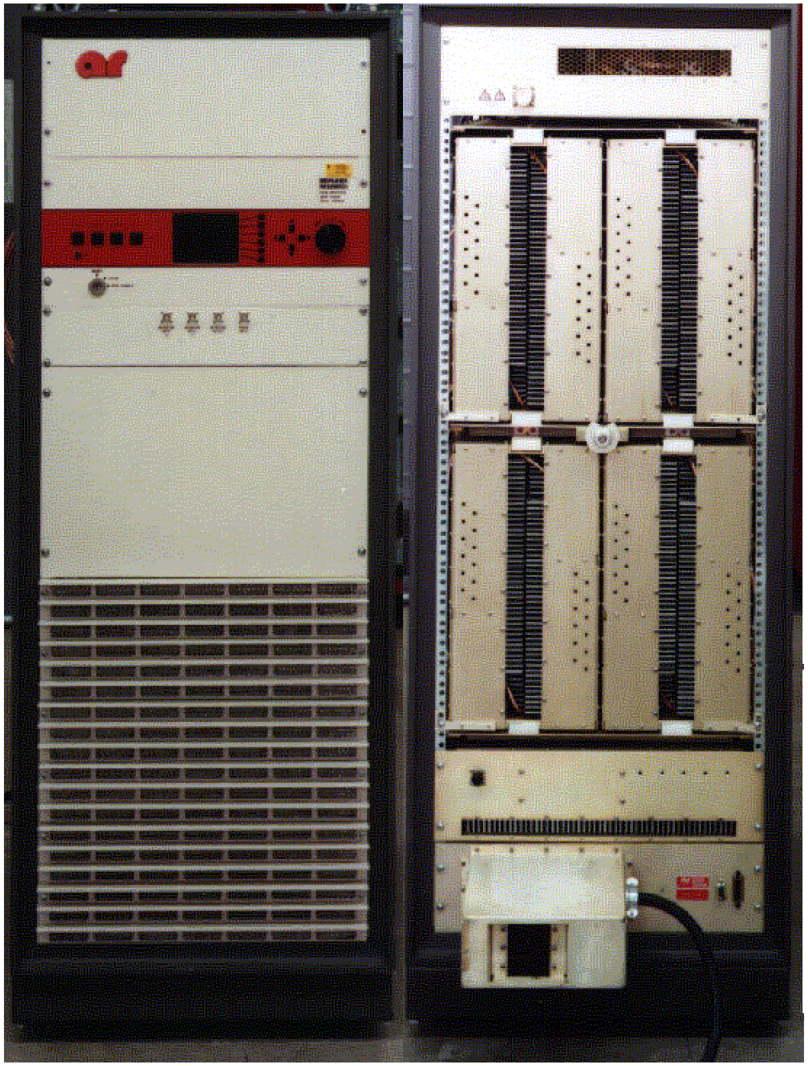


Figure 2: Front and Rear views of Amplifier Research model 3500A100.

- Recycler has four of these amps, capable of generating +/-2000V or arbitrary waveform.
- MI (D. Wildman) ordered 3 more for longitudinal Dampers, due ~May.

✍ ~1800V of broadband voltage in MI

Pbars vs. Proton Timing: Longitudinal

- 3 Cavites spanning 5-10 meters
- Bunch-by-bunch kick needs separate fanout for Protons and Pbars
- Either:
  - One DAC per Cavity
  - Relay switch box with different cable delays ✍ *this option chosen* ✍ single TTL bit Pbar-P

"Universal-Damper" Application: Signal Processing Steps (transverse)

Echotek Board
1) Bandwidth-Limit input signal to ~53 MHz
2) 14 Bit Digitization at 106 MHz or 212 MHz

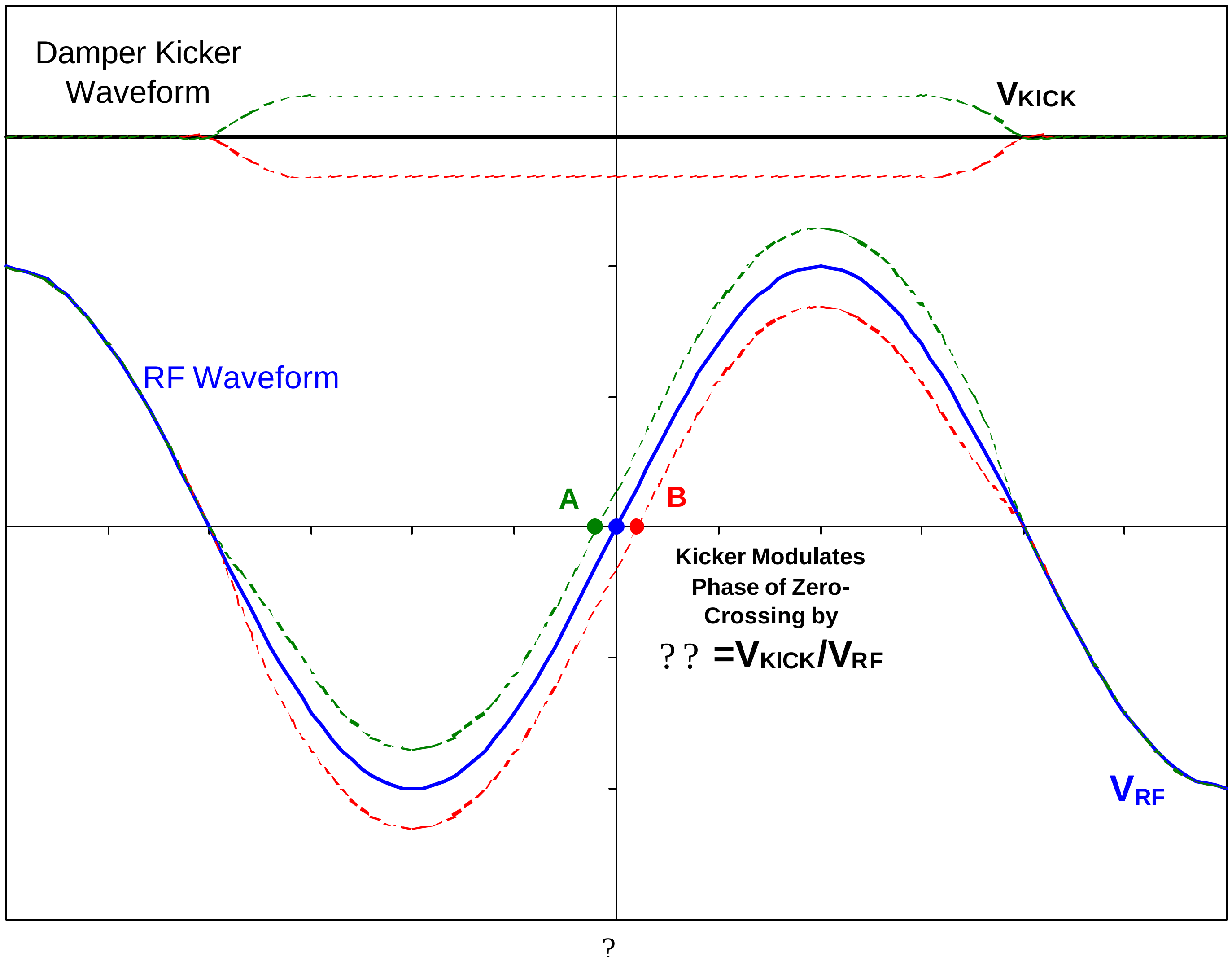
Inside FPGA
3) FIR filter to get single-bunch signal
4) Sum & Difference of plate signals
5) Multi turn difference filter (FIR) w/delay
6) Pickup Mixing for correct Betatron Phase
7) Bunch-by-bunch gain, dead band etc.
8) Timing Corrections for Frequency Sweep
9) Pre-Distortion for Kicker Power Amp

Buy
10) Power Amp for Kicker

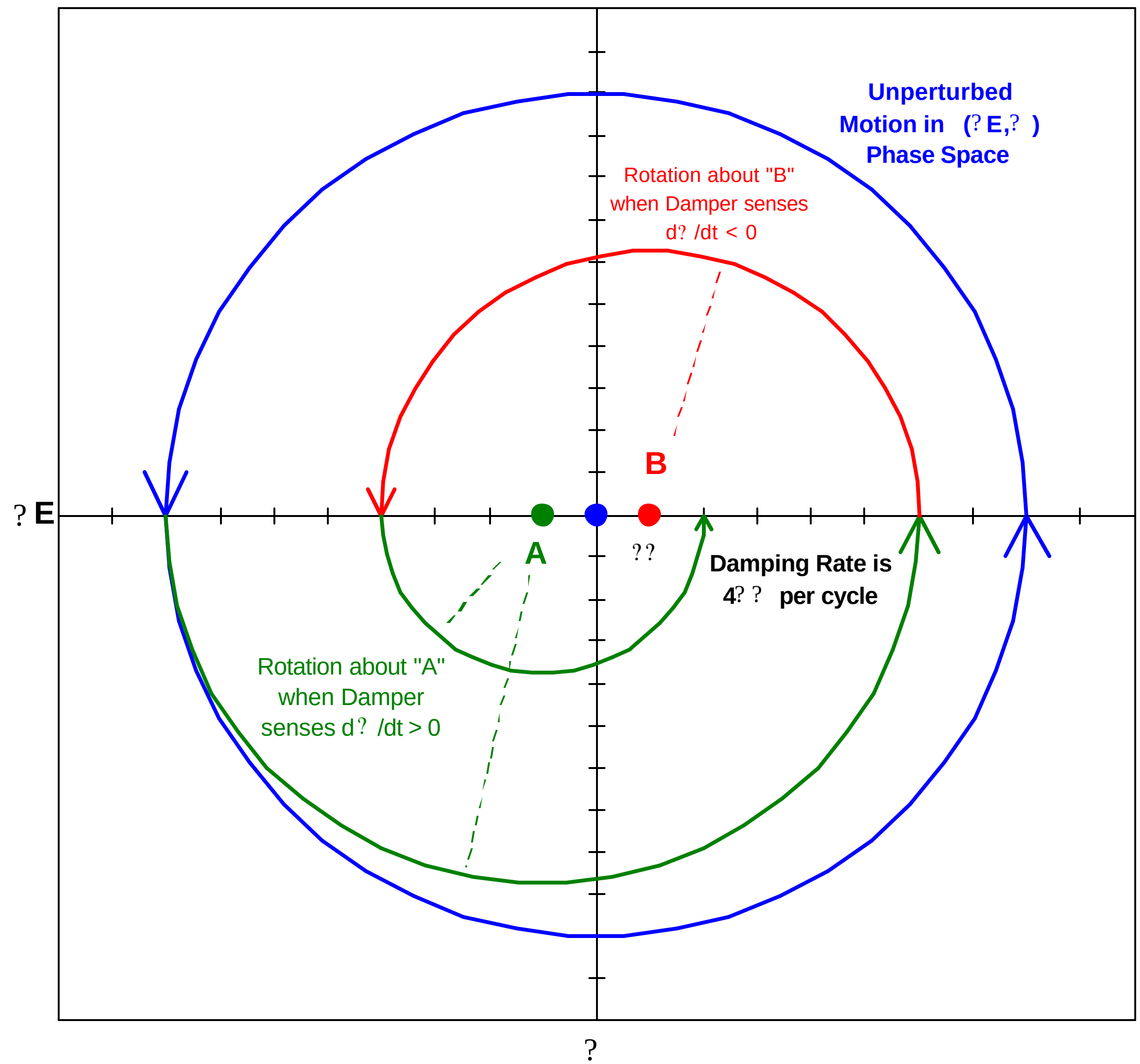
1. Longitudinal Damper in Main Injector

1. Benefits to Bunch Coalescing for Collider
   - "Dancing Bunches" degrade Proton coalescing and $?_L$
   - Affects Lum directly (hourglass) and indirectly (lifetime)
   - We are deliberately blowing $?_L$ in Booster
2. Benefits for Pbar Stacking Cycles
   - Bunch Rotation is generally turned off ! (x1.5 stack rate?)
   - Slip-Stacking etc. (Run IIb) will require stable bunches
3. Needed for eventual NUMI operation

Longitudinal Damper Works by Modulating Phase of RF Zero Crossing



Damping of Bunch Motion by Modulation of Center of Rotation (RF zero-crossing) on Alternate Half-cycles of Synchrotron Motion



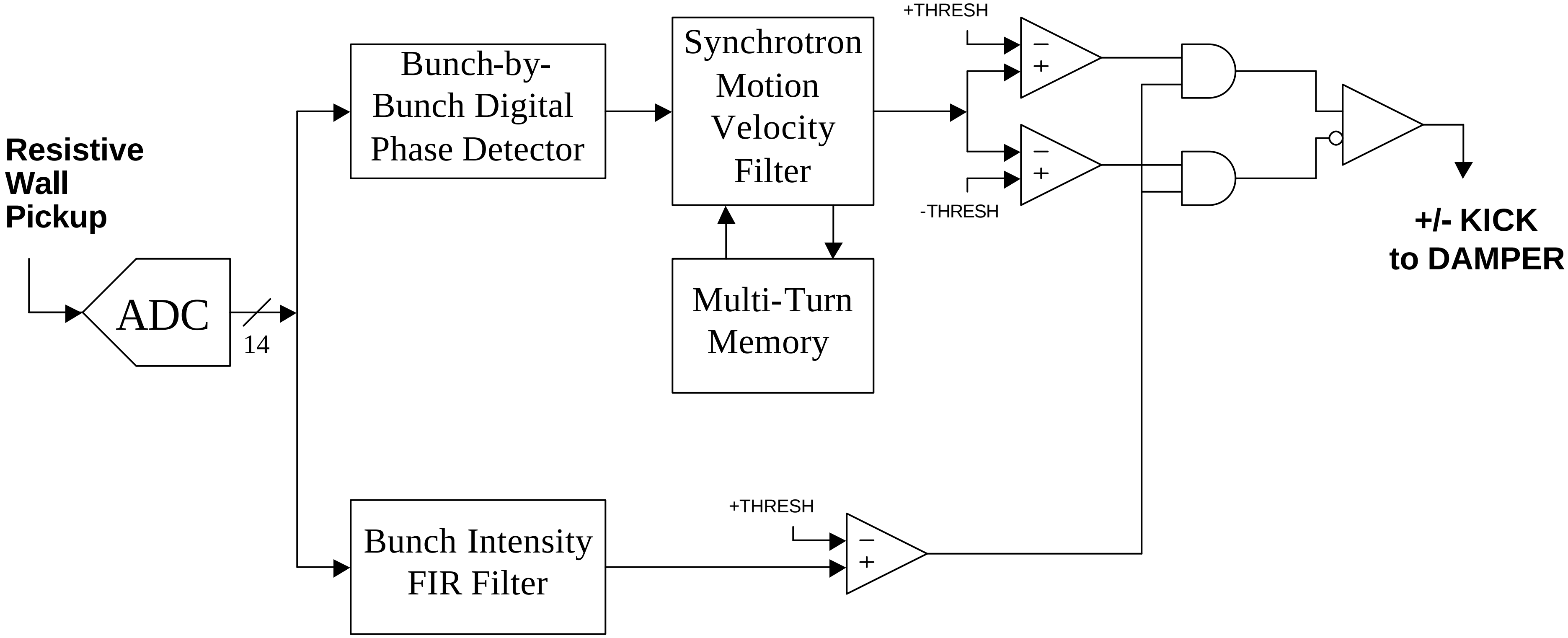
Numerical Examples for Longitudinal Dampers

	MI at Injection	Recycler
RF Voltage	1000 kV	2 kV
Damper Voltage	0.6 kV	0.1 kV
RF frequency	53 MHz	2.5 MHz
Sychrotron Freq.	870 Hz	8.5 Hz
Damping Time for 20 degree phase osc.	145 periods	1.7 periods
	0.17 sec.	0.21 sec.

Damping can be made faster

by raising V_{DAMPER} and/or lowering V_{RF}

Longitudinal Damper FPGA Logic



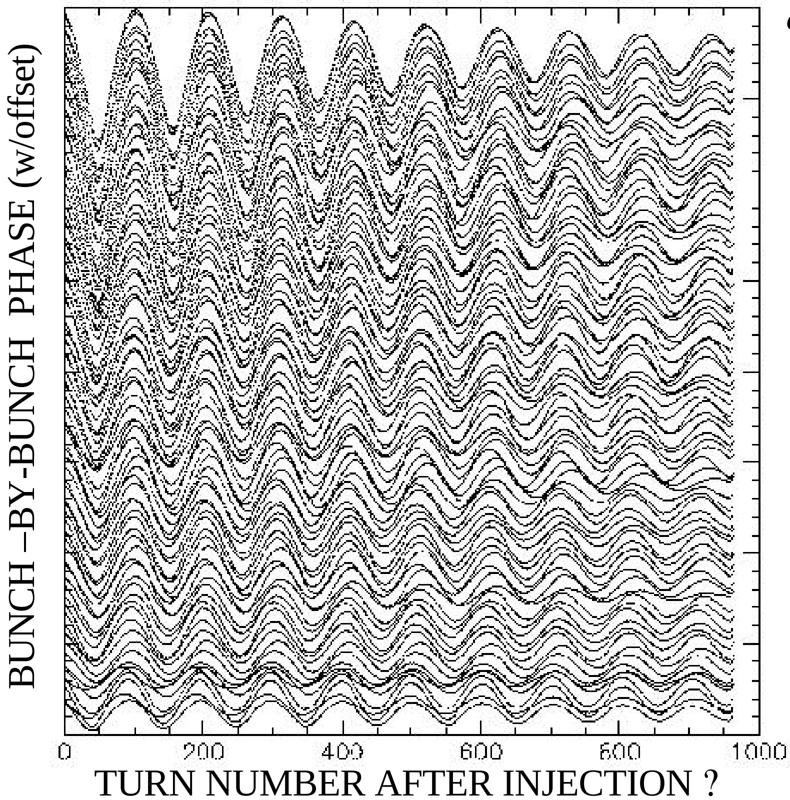
Individual Bunches are kicked + or – depending on whether they are moving right or left in phase

FPGA Code for Longitudinal Damper

```
---------------------------------------------------
--  Longitudinal Damper (53 MHz Bunch-by-Bunch)
---------------------------------------------------
-- Damper senses direction that bunch phase is changing, with RC lowpass filter to suppress noise.
-- RC filter takes a fraction of the difference between incoming sample and current value of filter,
--   and adds this fraction to the filter value for next time.  (The fraction determines filter time constant).
-- This procedure makes equilbrium value of filter equal to the incoming value, independent of time constant chosen.
-- This property is useful so that schmidt-trigger hysteresis value on filter output independent of time constant.
-- VME register selects time constants of 1,2,4 or 8 turns (corresponding to fraction=1,0.5,0.25 or 0.125)

-- bunch-by-bunch storage for filter output values (slightly less than 1 turn to allow for pipelined filter calcs)
()= delay_longfilt : altshift_taps()
    with (NUMBER_OF_TAPS=1, TAP_DISTANCE=RING_HARMONIC_NUMBER-1, WIDTH=16) returns (.taps[0]);

-- RC filter subtractor to take difference between incoming sample and current value of filter
()= sub_longfilt : lpm_add_sub() with (LPM_WIDTH=16, LPM_DIRECTION="sub") returns(.result[0]);
    sub_longfilt.dataa[]=Qch1.q[];              -- incoming sample: quadrature signal from Ch1 measures phase
    sub_longfilt.datab[]=delay_longfilt.shiftout[];      -- minus current filter value

-- arithmetic shifter to choose 1-turn, 2 turn, 4-turn, or 8-turn time constant
()= shifter_longfilt: lpm_clshift()
    with (LPM_WIDTH=16, LPM_WIDTHDIST=2, LPM_SHIFTTYPE="ARITHMETIC") returns (.result[0]);
    shifter_longfilt.data[]=sub_longfilt.result[];      -- input from subtractor (new value-filter value)
    shifter_longfilt.distance[1..0]=reg_DDXG12.q[1..0]; -- shift count controlled by register bits
    shifter_longfilt.direction=VCC;                     -- shift direction always to right (makes fraction 1,1/2..1/8)

-- Pipelined Adder to apply shifted difference to previous filter value to get longitudinal kick value
()= long_kick : lpm_add_sub() with (LPM_WIDTH=16, LPM_DIRECTION="add", LPM_PIPELINE=1) returns (.result[0]);
    long_kick.dataa[]=shifter_longfilt.result[];        -- add fraction from shifter
    long_kick.datab[]=delay_longfilt.shiftout[];    -- filter value from last time
    long_kick.clock = adclkby2;

-- Return filter value to shift register for next time around
    delay_longfilt.shiftin[]=long_kick.result[];
    delay_longfilt.clock=adclkby2;
    Longitudinal_Kick[] = long_kick.result[];
```

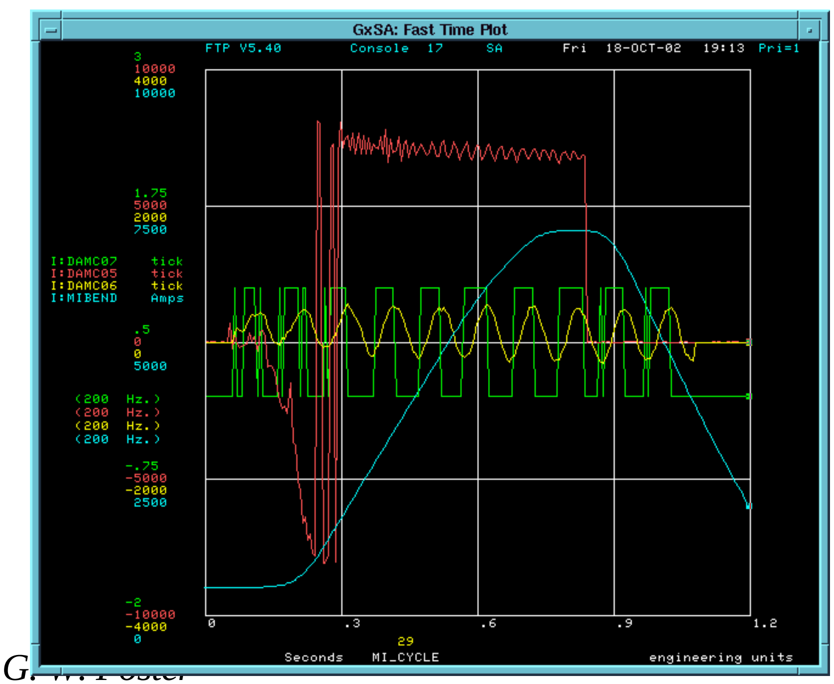
MI Longitudinal Damper

(Ashmanskas, Foster)



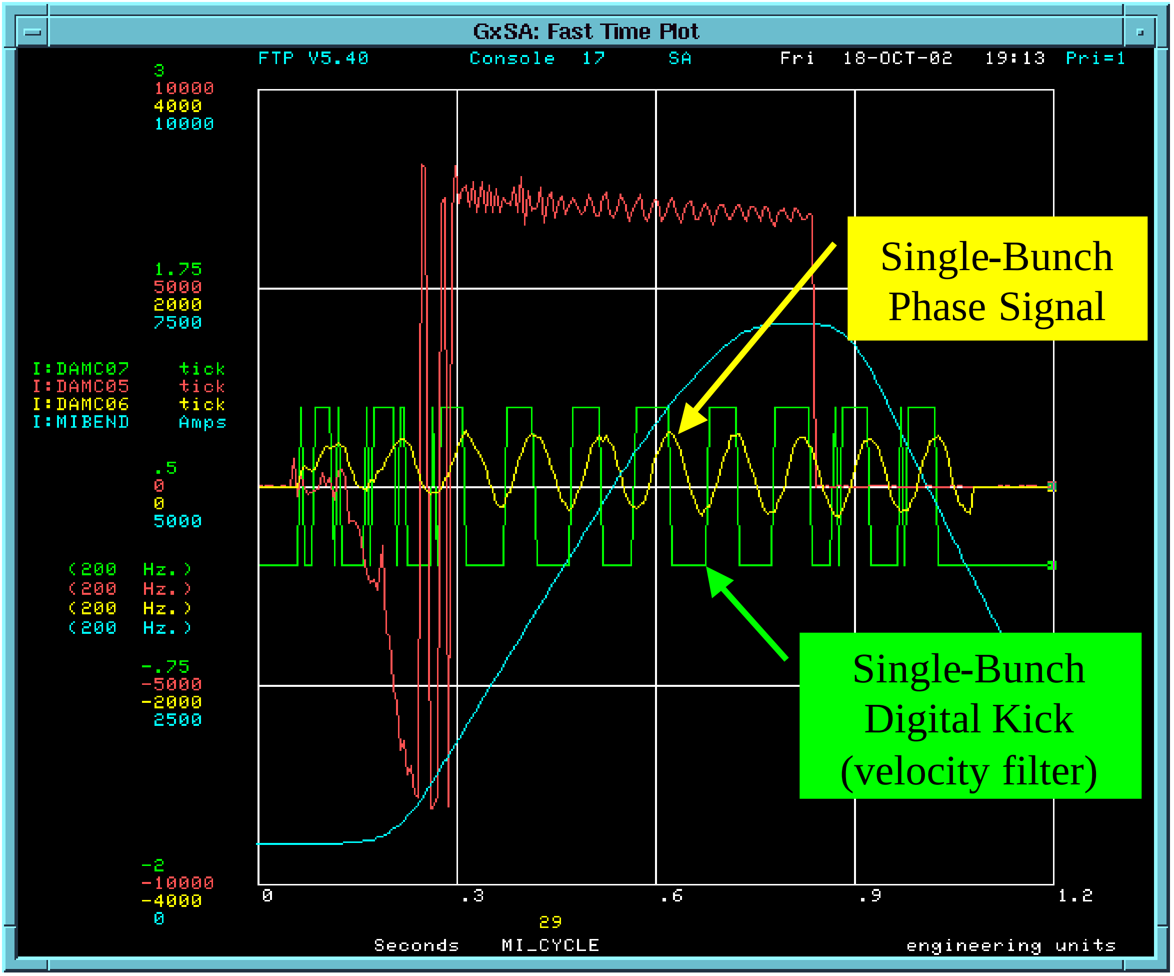
- ? 80 Bunch-by-Bunch synchrotron oscillations (on Pbar Stacking Cycle) measured with Echotek board & custom firmware

- Single Bunch Digital Kick ? using Digital Velocity Filter implemented in FPGA firmware



MI Longitudinal Damper Kick Calculated in FPGA Firmware

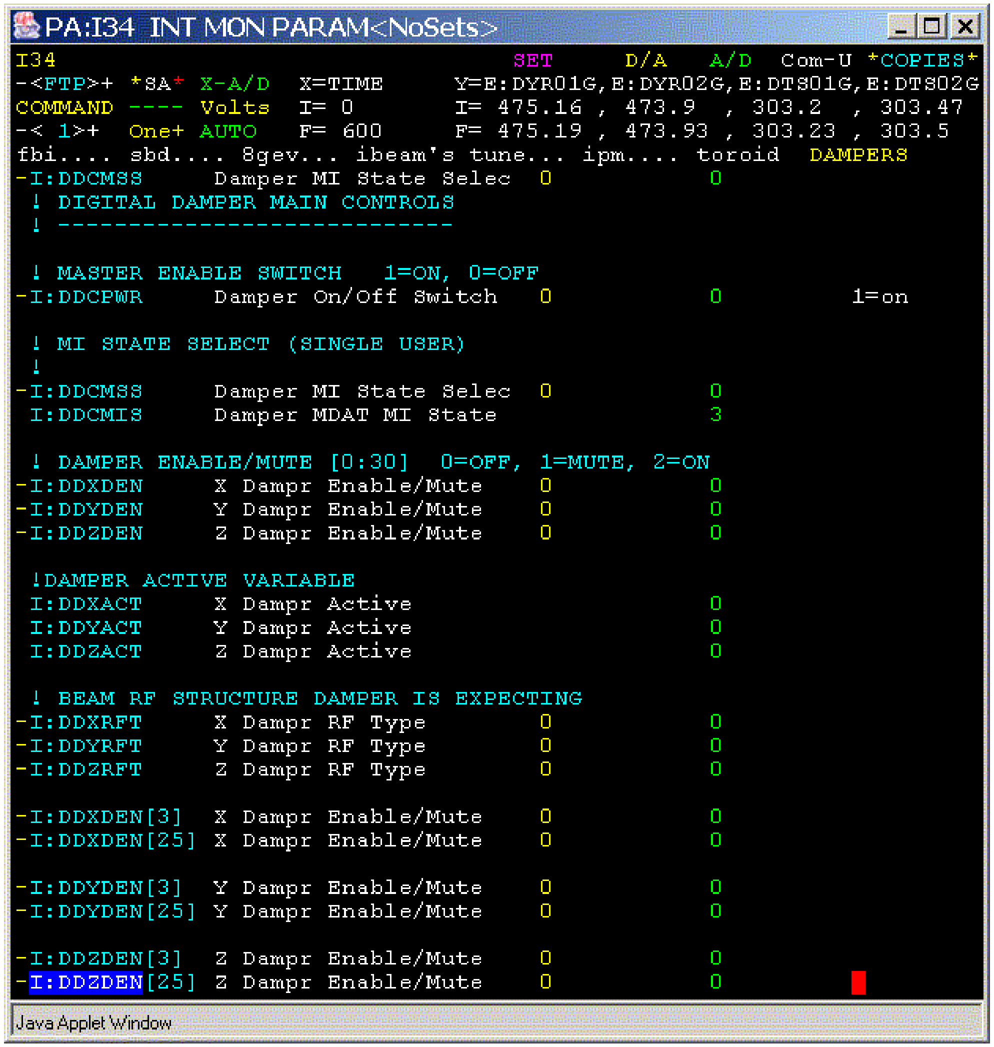
(<u>Ashmanskas</u>, Foster)



ACNET Issues

- Damper must behave differently for different bunches ✍ bunch-by-bunch RAM
  - Specifies Damper Gain, anti damp, noise injection, pinging, etc. on bunch-by-bunch basis.
- Damper must behave differently on different MI cycles
  - Each control register becomes an ACNET Array Device indexed by RF State
  - Register contents switch automatically when MI State changes *(D. Nicklaus)*

ACNET Control Devices (>250 total)



- MasterControl Registers typically single devices
- Most control registers are array devices indexed by MI State

PA:I34 INT MON PARAM<NoSets>

```
I34                                        SET      D/A    A/D   Com-U *COPIES*
-<FTP>+ *SA* X-A/D  X=TIME     Y=E:DYR01G,E:DYR02G,E:DTS01G,E:DTS02G
COMMAND ---- Volts  I= 0       I= 475.16 , 473.9  , 303.2  , 303.47
-< 2>+  One+ AUTO   F= 600     F= 475.19 , 473.93 , 303.23 , 303.5
fbi.... sbd.... 8gev... ibeam's tune... ipm.... toroid  DAMPERS
-I:DDCMSS      Damper MI State Selec  0           0
 !DAMPER THRESHOLDS AND DIAGNOSTICS
 !---------------------------------

 !THRESHOLD FOR 'BEAM PRESENT' ON RWM [0:30]
-I:DDZTHP      Z Dampr Thresh Beam P  0           0

 ! THRESHOLD FOR BEAM KICKED [0:30]
-I:DDXTHK      X Dampr Thresh to Kic  0           0
-I:DDYTHK      Y Dampr Thresh to Kic  0           0
-I:DDZTHK      Z Dampr Thresh to Kic  0           0

 !NUMBER OF BUNCHES PRESENT ABOVE THRESHOLD
 I:DDXNPR      X Dampr Nbr Bunches P              0         Bnch
 I:DDYNPR      Y Dampr Nbr Bunches P              0         Bnch
 I:DDZNPR      Z Dampr Nbr Bunches P              0         Bnch


 ! NUMBER OF BUNCHES KICKED (INCL. PINGER)
 I:DDXNKI      X Dampr Nbr Bunches K              0         Bnch
 I:DDYNKI      Y Dampr Nbr Bunches K              0         Bnch
 I:DDZNKI      Z Dampr Nbr Bunches K              0         Bnch
```

Java Applet Window

PA:I34 INT MON PARAM<NoSets>

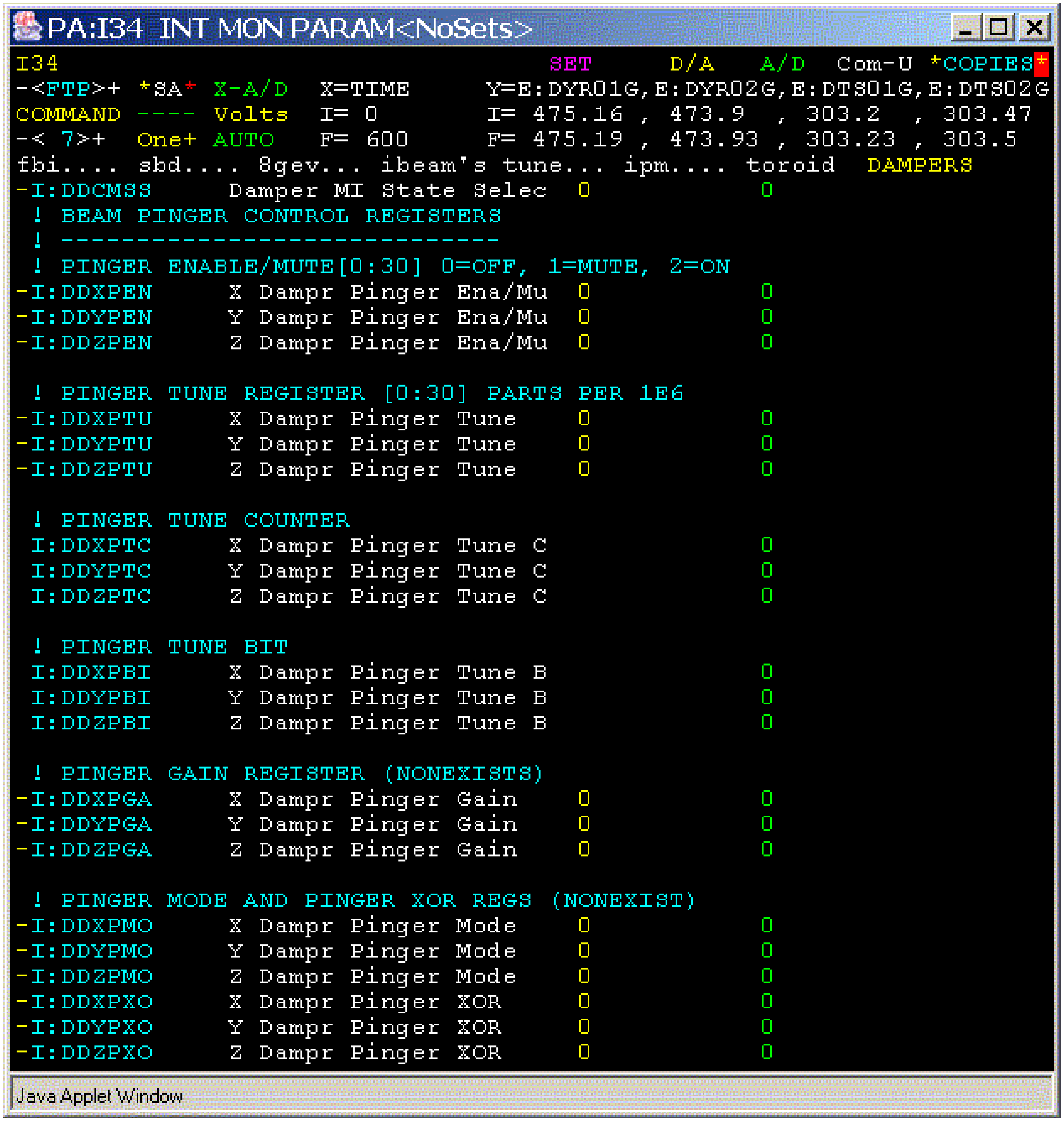
```
I34                                           SET       D/A    A/D   Com-U  *COPIES*
-<FTP>+ *SA* X-A/D  X=TIME     Y=E:DYR01G,E:DYR02G,E:DTS01G,E:DTS02G
COMMAND ----  Volts  I= 0       I= 475.16 , 473.9  , 303.2  , 303.47
-< 5>+  One+ AUTO   F= 600     F= 475.19 , 473.93 , 303.23 , 303.5
fbi.... sbd.... 8gev... ibeam's tune... ipm.... toroid  DAMPERS
-I:DDCMSS     Damper MI State Selec  0           0
 ! DAMPER TIMER DEVICES
 ! --------------------

 ! TIME BASE (NONEXISTS)
-I:DDXTBA     X Dampr Time Base      0           0
-I:DDYTBA     Y Dampr Time Base      0           0
-I:DDZTBA     Z Dampr Time Base      0           0

 !DAMPER FIRST TURN [0:30] 21 BITS
-I:DDXD1T     X Dampr 1st Turn Acti  0           0         Turn
-I:DDYD1T     Y Dampr 1st Turn Acti  0           0         Turn
-I:DDZD1T     Z Dampr 1st Turn Acti  0           0         Turn

 !DAMPER LENGTH IN TURNS [0:30] 21 BITS
-I:DDXDLT     X Dampr Length in Tur  0           0         Turn
-I:DDYDLT     Y Dampr Length in Tur  0           0         Turn
-I:DDZDLT     Z Dampr Length in Tur  0           0         Turn

 ! DAMPER TURNCOUNTER SINCE MI RESET (NEEDS DDXBK0)
 I:DDXTCR     X Dampr Turns since R              2         Turn
 I:DDYTCR     Y Dampr Turns since R              223076    Turn
 I:DDZTCR     Z Dampr Turns since R              223076    Turn

 ! DAMPER TURNS ACTIVE COUNTER
 I:DDXDTA     X Dampr Active Turns               0         Turn
 I:DDYDTA     Y Dampr Active Turns               0         Turn
 I:DDZDTA     Z Dampr Active Turns               0         Turn

 !NUMBER OF MI RESETS
-I:DDXNMR     X Dampr MI  19143     19156       19156
-I:DDYNMR     Y Dampr MI  19144     19157       19157
-I:DDZNMR     Z Dampr MI  19133     19146       19146
```

Java Applet Window

PA:I34 INT MON PARAM<NoSets>

```
I34                                         SET       D/A    A/D   Com-U  *COPIES*
-<FTP>+  *SA*  X-A/D   X=TIME     Y=E:DYR01G,E:DYR02G,E:DTS01G,E:DTS02G
COMMAND  ----  Volts   I= 0       I= 475.16 , 473.9   , 303.2   , 303.47
-<  6>+   One+ AUTO    F= 600     F= 475.19 , 473.93  , 303.23  , 303.5
fbi.... sbd.... 8gev... ibeam's tune... ipm.... toroid  DAMPERS
-I:DDCMSS      Damper MI State Selec  0           0
 ! DAMPER KICK TIMING
 ! ------------------

 ! BUCKET 0 POSITION ADJUST [0:30]
-I:DDXBK0      X Dampr Bucket 0 Pos.  0           0         Bkt.
-I:DDYBK0      Y Dampr Bucket 0 Pos.  0           0         Bkt.
-I:DDZBK0      Z Dampr Bucket 0 Pos.  0           0         Bkt.

 ! KICK DELAY REGISTER [0:30]
-I:DDXKDL      X Dampr Kick Delay     0           0         Bkt.
-I:DDYKDL      Y Dampr Kick Delay     0           0         Bkt.
-I:DDZKDL      Z Dampr Kick Delay     0           0         Bkt.

 ! MOMENTUM-DEPENDENT KICK DELAY LOOKUP (NONEXISTS)
-I:DDXKDR      X Dampr Kick Delay Ta  0           0
-I:DDXKDR[1]   X Dampr Kick Delay Ta  0           0
-I:DDYKDR      Y Dampr Kick Delay Ta  0           0
-I:DDYKDR[1]   Y Dampr Kick Delay Ta  0           0
-I:DDZKDR      Z Dampr Kick Delay Ta  0           0
-I:DDZKDR[1]   Z Dampr Kick Delay Ta  0           0
 ! KICKER OUTPUT PREEMPHASIS FILTER COEF.(NONEXISTS
-I:DDXPEC      X Dampr PreEmphasis C  0           0
-I:DDXPEC[1]   X Dampr PreEmphasis C  0           0
-I:DDXPEC[2]   X Dampr PreEmphasis C  0           0
-I:DDXPEC[3]   X Dampr PreEmphasis C  0           0

-I:DDYPEC      Y Dampr PreEmphasis C  0           0
-I:DDYPEC[1]   Y Dampr PreEmphasis C  0           0
-I:DDYPEC[2]   Y Dampr PreEmphasis C  0           0
-I:DDYPEC[3]   Y Dampr PreEmphasis C  0           0
-I:DDYPEC[4]   Y Dampr PreEmphasis C  0           0

-I:DDZPEC      Z Dampr PreEmphasis C  0           0
-I:DDZPEC[1]   Z Dampr PreEmphasis C  0           0
```

Java Applet Window

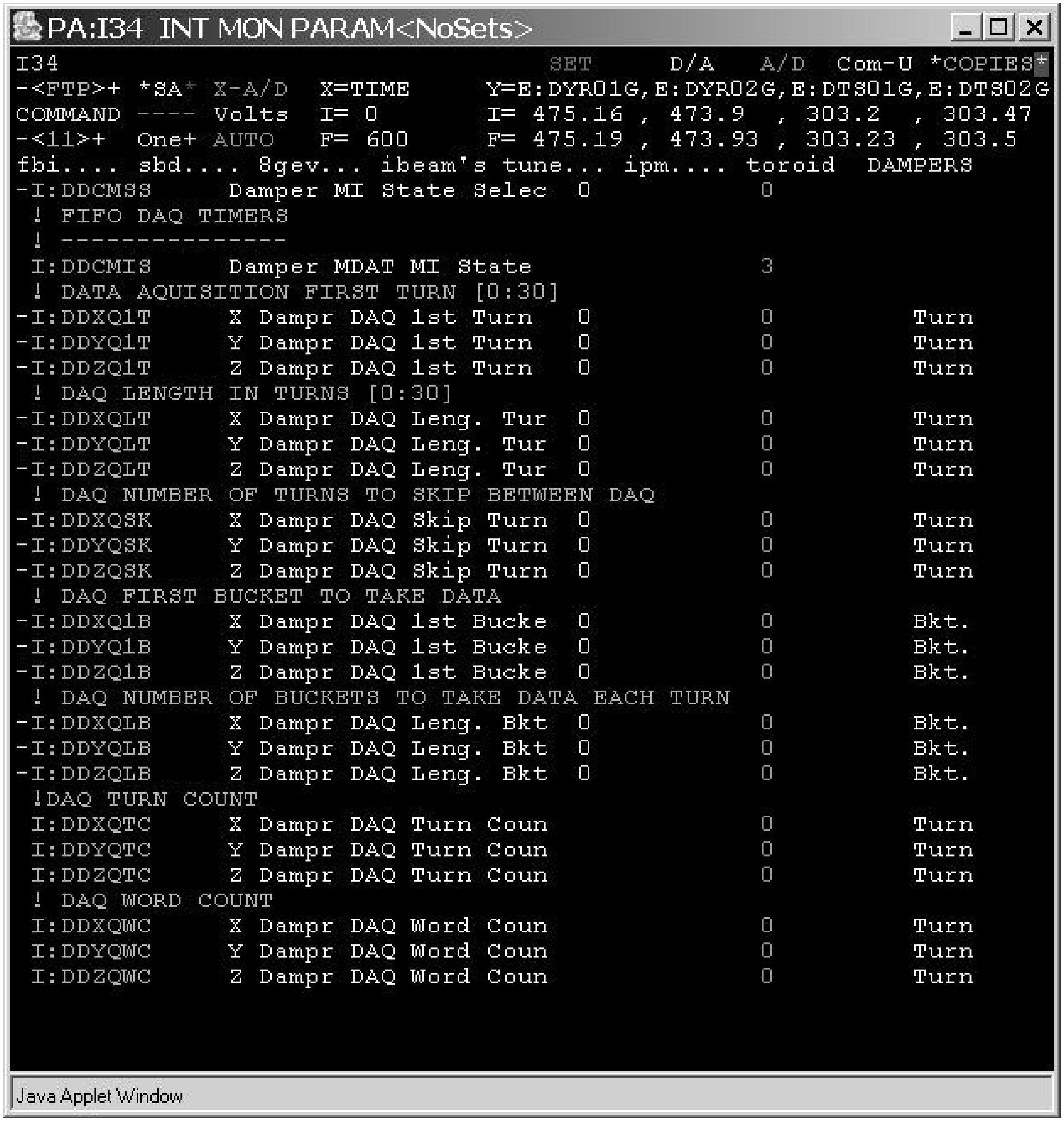

PA:I34 INT MON PARAM<NoSets>
I34 SET D/A A/D Com-U *COPIES*
-<FTP>+ *SA* X-A/D X=TIME Y=E:DYR01G,E:DYR02G,E:DTS01G,E:DTS02G
COMMAND ---- Volts I= 0 I= 475.16 , 473.9 , 303.2 , 303.47
-< 7>+ One+ AUTO F= 600 F= 475.19 , 473.93 , 303.23 , 303.5
fbi.... sbd.... 8gev... ibeam's tune... ipm.... toroid DAMPERS
-I:DDCMSS Damper MI State Selec 0 0
! BEAM PINGER CONTROL REGISTERS
! ----------------------------
! PINGER ENABLE/MUTE[0:30] 0=OFF, 1=MUTE, 2=ON
-I:DDXPEN X Dampr Pinger Ena/Mu 0 0
-I:DDYPEN Y Dampr Pinger Ena/Mu 0 0
-I:DDZPEN Z Dampr Pinger Ena/Mu 0 0
! PINGER TUNE REGISTER [0:30] PARTS PER 1E6
-I:DDXPTU X Dampr Pinger Tune 0 0
-I:DDYPTU Y Dampr Pinger Tune 0 0
-I:DDZPTU Z Dampr Pinger Tune 0 0
! PINGER TUNE COUNTER
I:DDXPTC X Dampr Pinger Tune C 0
I:DDYPTC Y Dampr Pinger Tune C 0
I:DDZPTC Z Dampr Pinger Tune C 0
! PINGER TUNE BIT
I:DDXPBI X Dampr Pinger Tune B 0
I:DDYPBI Y Dampr Pinger Tune B 0
I:DDZPBI Z Dampr Pinger Tune B 0
! PINGER GAIN REGISTER (NONEXISTS)
-I:DDXPGA X Dampr Pinger Gain 0 0
-I:DDYPGA Y Dampr Pinger Gain 0 0
-I:DDZPGA Z Dampr Pinger Gain 0 0
! PINGER MODE AND PINGER XOR REGS (NONEXIST)
-I:DDXPMO X Dampr Pinger Mode 0 0
-I:DDYPMO Y Dampr Pinger Mode 0 0
-I:DDZPMO Z Dampr Pinger Mode 0 0
-I:DDXPXO X Dampr Pinger XOR 0 0
-I:DDYPXO Y Dampr Pinger XOR 0 0
-I:DDZPXO Z Dampr Pinger XOR 0 0
Java Applet Window

PA:I34 INT MON PARAM<NoSets>

```
I34                                            SET       D/A    A/D    Com-U *COPIES*
-<FTP>+ *SA* X-A/D  X=TIME      Y=E:DYR01G,E:DYR02G,E:DTS01G,E:DTS02G
COMMAND ---- Volts  I= 0        I= 475.16 , 473.9   , 303.2   , 303.47
-< 8>+  One+ AUTO   F= 600      F= 475.19 , 473.93  , 303.23  , 303.5
fbi.... sbd.... 8gev... ibeam's tune... ipm.... toroid  DAMPERS
-I:DDCMSS     Damper MI State Selec  0            0
 ! BEAM PINGER TIMER REGISTERS

 ! PINGER ACTIVE SIGNAL
 I:DDXPAC     X Dampr Pinger Active               0
 I:DDYPAC     Y Dampr Pinger Active               0
 I:DDZPAC     Z Dampr Pinger Active               0

 ! PINGER 1ST TURN TO ACTIATE
-I:DDXP1T     X Dampr Pinger 1st Tu  0            0
-I:DDYP1T     Y Dampr Pinger 1st Tu  0            0
-I:DDZP1T     Z Dampr Pinger 1st Tu  0            0

 ! PINGER LENGTH IN TURNS TO STAY ACTIVE
-I:DDXPLT     X Dampr Pinger Len.Tu  0            0
-I:DDYPLT     Y Dampr Pinger Len.Tu  0            0
-I:DDZPLT     Z Dampr Pinger Len.Tu  0            0

 !PINGER 1ST BUCKET TO HIT
-I:DDXP1B     X Dampr Pinger 1st Bk  0            0
-I:DDYP1B     Y Dampr Pinger 1st Bk  0            0
-I:DDZP1B     Z Dampr Pinger 1st Bk  0            0

 ! PINGER LENGTH IN BUCKETS TO HIT
-I:DDXPLB     X Dampr Pinger Leng.   0            0
-I:DDYPLB     Y Dampr Pinger Leng.   0            0
-I:DDZPLB     Z Dampr Pinger Leng.   0            0

 ! PINGER ACTIVE TURNCOUNTER
 I:DDXPAT     X Dampr Pinger Turnco               0
 I:DDYPAT     Y Dampr Pinger Turnco               0
 I:DDZPAT     Z Dampr Pinger Turnco               0
```

Java Applet Window

```
PA:I34  INT MON PARAM<NoSets>
I34   MI60  2.5MHZ  BPMS                   SET        D/A     A/D    Com-U  *COPIES*
-<FTP>+  *SA*  X-A/D    X=TIME       Y=E:DYR01G,E:DYR02G,E:DTS01G,E:DTS02G
COMMAND  ----  Volts    I= 0         I= 475.16 , 473.9   , 303.2   , 303.47
-<10>+   One+  AUTO     F= 600       F= 475.19 , 473.93  , 303.23  , 303.5
fbi....  sbd....  8gev...  ibeam's  tune...  ipm....  toroid   DAMPERS
-I:DDCMSS        Damper MI State Selec  0            0
 ! FIFO DAQ CONTROL REGISTERS
 ! -------------------------

 ! DAQ REQUEST REGISTER [0:30]  1-BIT
-I:DDXQRQ        X Dampr DAQ Request B  0            0
-I:DDYQRQ        Y Dampr DAQ Request B  0            0
-I:DDZQRQ        Z Dampr DAQ Request B  0            0

 ! DAQ REQUEST STATUS REG: 0=IDLE,1=PENDING,2=ACTIV
 I:DDXQRS        X Dampr DAQ Req. Stat               0
 I:DDYQRS        Y Dampr DAQ Req. Stat               0
 I:DDZQRS        Z Dampr DAQ Req. Stat               0

 ! DAQ FIFO#0: NOT A GOOD IDEA TO PUT THESE ON PAGE
 !I:DDXQF0        X Dampr DAQ FIFO 0    -1           -1
 !I:DDYQF0        Y Dampr DAQ FIFO 0    -1           -1
 !I:DDZQF0        Z Dampr DAQ FIFO 0    -1           -1

 ! FIFO #1
 !I:DDXQF1        X Dampr DAQ FIFO 1    -1           -1
 !I:DDYQF1        Y Dampr DAQ FIFO 1    -1           -1
 !I:DDZQF1        Z Dampr DAQ FIFO 1    -1           -1

 ! DAQ MUX CONTROL REGISTER [0:30]
-I:DDXQM1        X Dampr DAQ Mux        0            0
-I:DDYQM1        Y Dampr DAQ Mux        0            0
-I:DDZQM1        Z Dampr DAQ Mux        0            0

Java Applet Window
```


PA:I34 INT MON PARAM<NoSets>
I34 SET D/A A/D Com-U *COPIES*
-<FTP>+ *SA* X-A/D X=TIME Y=E:DYR01G,E:DYR02G,E:DTS01G,E:DTS02G
COMMAND ---- Volts I= 0 I= 475.16 , 473.9 , 303.2 , 303.47
-<11>+ One+ AUTO F= 600 F= 475.19 , 473.93 , 303.23 , 303.5
fbi.... sbd.... 8gev... ibeam's tune... ipm.... toroid DAMPERS
-I:DDCMSS Damper MI State Selec 0 0
! FIFO DAQ TIMERS
! ---------------
I:DDCMIS Damper MDAT MI State 3
! DATA AQUISITION FIRST TURN [0:30]
-I:DDXQ1T X Dampr DAQ 1st Turn 0 0 Turn
-I:DDYQ1T Y Dampr DAQ 1st Turn 0 0 Turn
-I:DDZQ1T Z Dampr DAQ 1st Turn 0 0 Turn
! DAQ LENGTH IN TURNS [0:30]
-I:DDXQLT X Dampr DAQ Leng. Tur 0 0 Turn
-I:DDYQLT Y Dampr DAQ Leng. Tur 0 0 Turn
-I:DDZQLT Z Dampr DAQ Leng. Tur 0 0 Turn
! DAQ NUMBER OF TURNS TO SKIP BETWEEN DAQ
-I:DDXQSK X Dampr DAQ Skip Turn 0 0 Turn
-I:DDYQSK Y Dampr DAQ Skip Turn 0 0 Turn
-I:DDZQSK Z Dampr DAQ Skip Turn 0 0 Turn
! DAQ FIRST BUCKET TO TAKE DATA
-I:DDXQ1B X Dampr DAQ 1st Bucke 0 0 Bkt.
-I:DDYQ1B Y Dampr DAQ 1st Bucke 0 0 Bkt.
-I:DDZQ1B Z Dampr DAQ 1st Bucke 0 0 Bkt.
! DAQ NUMBER OF BUCKETS TO TAKE DATA EACH TURN
-I:DDXQLB X Dampr DAQ Leng. Bkt 0 0 Bkt.
-I:DDYQLB Y Dampr DAQ Leng. Bkt 0 0 Bkt.
-I:DDZQLB Z Dampr DAQ Leng. Bkt 0 0 Bkt.
!DAQ TURN COUNT
I:DDXQTC X Dampr DAQ Turn Coun 0 Turn
I:DDYQTC Y Dampr DAQ Turn Coun 0 Turn
I:DDZQTC Z Dampr DAQ Turn Coun 0 Turn
! DAQ WORD COUNT
I:DDXQWC X Dampr DAQ Word Coun 0 Turn
I:DDYQWC Y Dampr DAQ Word Coun 0 Turn
I:DDZQWC Z Dampr DAQ Word Coun 0 Turn
Java Applet Window

What ADC Clock Speed is needed?

- ~53 MHz Bandwidth limited signal, sampled by 106 MHz ADC, measures either *in-phase* (cosine) or *quadrature* (sine) component
  - but not both ==> ADC clock phasing matters!
- 212 MHz sampling measures both *in-phase* and *quadrature* components. *Phasing is not critical* to determine vector magnitude.
- 212 MHz ✍ built in phase measurement

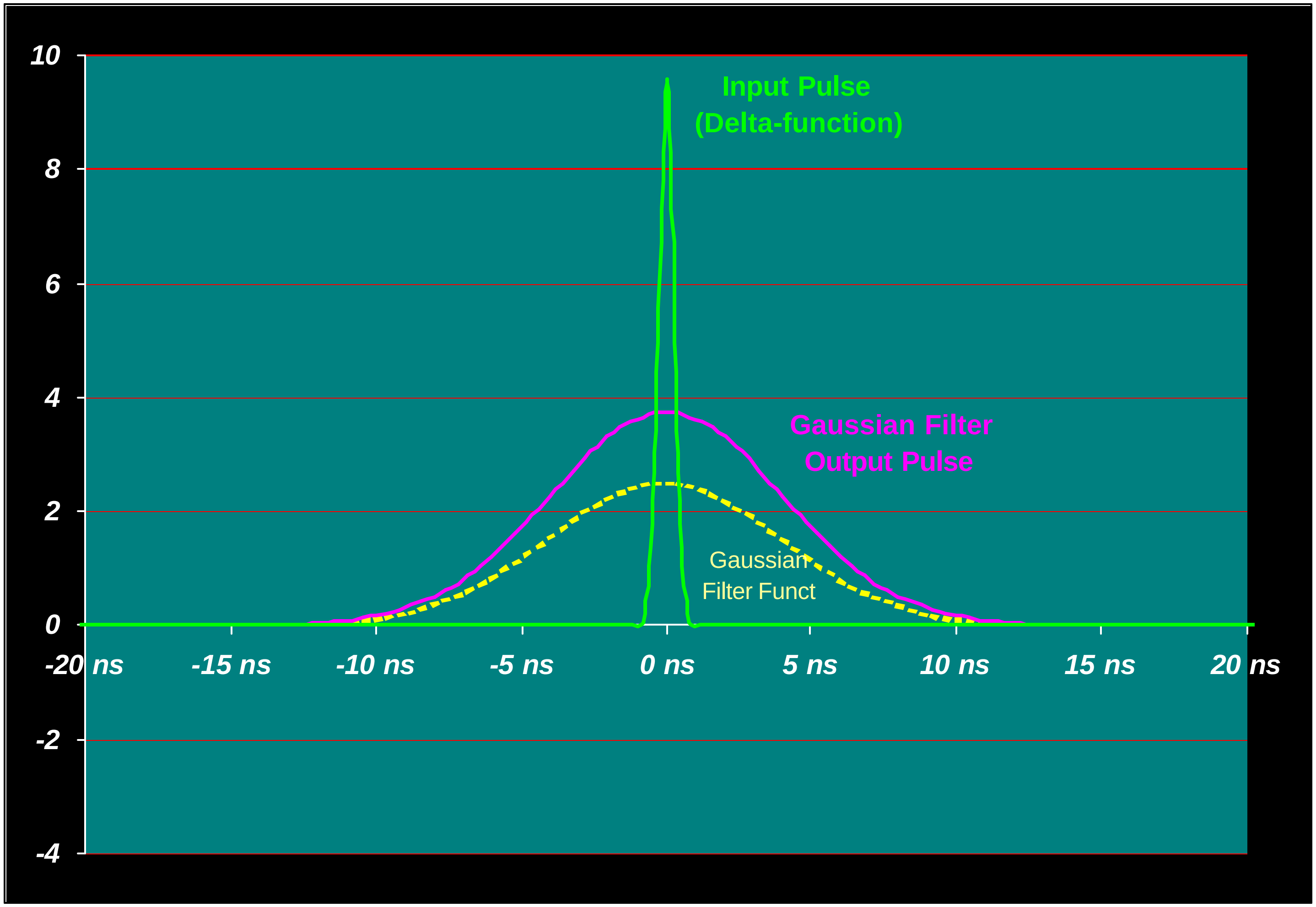
Bandwidth Limit Signal

- Raw signal has high-frequency components which can cause signal to be missed by ADC
  - "Aliasing"

- Bandwidth limited signal (to ~50 MHz) cannot be missed by 106 MHz ADC

- Eliminate low-frequency ripple, baseline shifts, etc. with Transformer or AC coupling
  - Digital Filtering can provide additional rejection

Gaussian Filter - Impulse Response

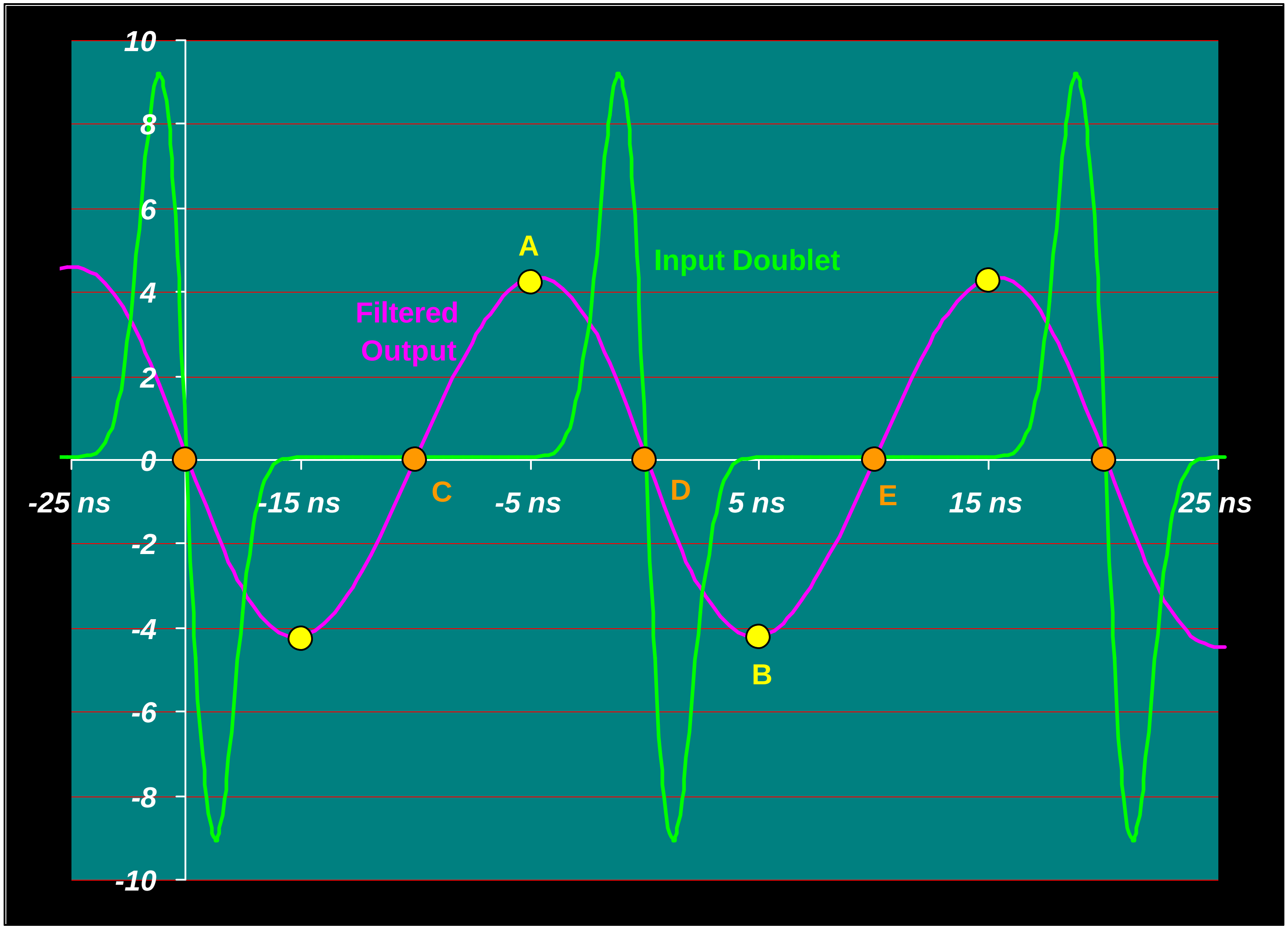


Spreads signal +/-5ns in time so it will not be missed by ADC

Reduces ADC Dynamic Range requirement, since spike does not have to be digitized

- Many implementations, e.g. traversal filter

In-Phase and Quadrature Sampling



"A - B" gives bunch-by-bunch "in-phase" signal

"D - (C+E)/2" gives bunch-by-bunch "out-of-phase" or "quadrature" signal

Vector Sum sqrt(I**2 +Q**2) is insensitive to clock jitter

- This is the argument for sampling at 2x Nyquist